Hinweis zur Abgabe:

Erstellen Sie ein Dokument mit den Ergebnissen der Aufgaben. Abgabe als pdf Dokument.

1. Gruppe auf ILIAS erstellen: **maximal 4 Studierende**
2. Kopfzeile der Seiten des Dokuments: Ihre Gruppennummer (ILIAS)
3. Erstellen Sie auf der ersten Seite des Dokuments eine Tabelle mit den Namen, Immatrikulationsnummern und Studiengang der Gruppenmitglieder.
   Vorlage

Name	**Vorname**	**Matrikelnummer**	**Studiengang**

A) **Abgabe:** Laden Sie das **pdf Dokument** mit der Lösung **und** ein **zip oder tar.gz/tgz Archiv** (keine anderen Formate!!) mit dem vollständigen Quellcode (umfasst auch Makefile/Startdateien) und Ergebnisdateien auf ILIAS hoch. KEINE BINARIES!!!
⇨ **2 Dateien !**

B) Spätester Zeitpunkt der Abgabe: 09. Jan 2023, 18:00 Uhr

Die beiden Aufgaben bauen aufeinander auf. Lösen Sie diese in der angegebenen Reihenfolge. Generell gilt, dass Sie Teile aus den Übungen übernehmen können.

Aufgabe 1: Rechnen mit komplexen Zahlen

Erstellen Sie ein Klasse mit dem Namen MyComplex zum Rechnen mit komplexen Zahlen. Implementieren Sie hierzu die notwendigen Methoden und Operatoren. Spalten Sie die Klasse in Kopf- und Quelldatei auf. Verwenden Sie zum Übersetzen ein Makefile. Sie können ein Makefile aus der Übung übernehmen und anpassen.

Folgende Rechenoperationen sind in Aufgabe 2 ggf. notwendig und müssen daher implementiert sein. Aufgabe 1 ist der Testfall für Ihre Klasse für:

- Addition(en), Subtraktion
- Multiplikation(en): `operator*`
- Zuweisungsoperator: `operator=`
- `operator-`
- Berechnung des Betrags einer komplexen Zahl: Methodenname `.norm()`
- Rückgabe von Realteil: Methodenname `.real()`
- Rückgabe des Imaginärteils: Methodenname `.imag()`
- `operator<<` Format: Ausgabe von real und imag Teil wie folgt => (x , y) wobei z=x+iy ist.

Beachten Sie bei der Implementierung die Konstantkorrektheit!

Fügen Sie den Quellcode in Ihre Abgabedatei (pdf-Datei) als Text ein; Sie können diesen aus eclipse (oder Ihrem Texteditor) herauskopieren oder unter LaTeX als Quellcode einbinden.

Übungsaufgabe mit Abgabe

Verwenden Sie das Testprogramm main_complex_beispiel.cpp zur Berechnung der folgenden Zahlenbeispiele: z_n sei eine komplexe Zahl $z = x + iy$ (x ist der Realteil und y ist der Imaginärteil).
Gegeben sind $z_1 = 2. +7.\ i$ und $z_2 = 42.\ \ -9. i$ und $z_3 = -11. +19.\ i.$

Berechnen Sie:

a) $z_4 = \ z_1 * z_2$
b) $z_5 = (z_1 + z_2)$
c) $z_6 = (z_1 + z_2) * 2.$
d) $z_7 = (z_2 + z_3) * z_1$
e) $z_8 = z_1 + 5.$
f) $z_9 = -z_1 + z_2$

Ergebnisdarstellung:

A) Listen Sie die Ausgabe / Ergebnisse der oben genannten Beispielaufgaben a) bis f) in der gegebenen Reihenfolge in ihrer pdf-Datei auf. (hier keine Anpassung des Quellcodes notwendig/erlaubt !!)

B) Schreiben Sie die Fälle a)-f) in der Operatorschreibweise im C++ Quellcode und fügen Sie diese Codezeilen in Ihre pdf-Datei ein!

Aufgabe 2: Untersuchung der Konvergenz komplexer Zahlenfolgen (Mandelbrot und Julia Mengen)

Das Konvergenzverhalten komplexer Zahlenfolgen soll untersucht werden. Erstellen Sie hierzu ein eigenes Programm, das die unten beschriebenen Eingabeparameter einliest, die unten gegebenen Iterationsvorschriften berechnet und jeweils eine Ausgabedatei im beschriebenen Format zurückgibt. Die Iterationsvorschrift ist vorgegeben (Beispiel $z_{i+1} = (z_i)^n + c$, wobei z_i und c komplexe Zahlen sind, n ein ganzzahliger Exponent) und i der Iterationsindex ist. Die Folge wird als konvergiert betrachtet, wenn der Betrag der Zahlen $|z_i| < R_c$ endlich (kleiner als ein vorgegebener Wert R_c) bleibt, wobei eine maximale Anzahl von Iterationsschritten $i \in [1, N_{max}]$ vorgeben wird.

Die Konvergenz der Folge soll im einem zweidimensionalen Wertebereich in der komplexen Koordinatenebene untersucht werden. Der Wertebereich ist durch Angabe der linken unteren Ecke (ll) und rechte obere Ecke (ur) festgelegt.
$\left(z_{ll} = z_{lower-left}\right), \dots, \left(z_{ur} = z_{upper-right}\right)$

Unterteilen Sie hierzu die Achse für den Realanteil und Imaginäranteil regelmäßig (Schrittweite δx und δy)

Darstellung des Ergebnisses mit Gnuplot:
Plotten Sie mit Hilfe der Befehle aus Übung 3 (Game of Life) ein „Bild“ mit dem Dateiformat.
`nx`: x-Richtung Nummer des diskretisierten Wertebereichs
`ny`: y-Richtung Nummer des diskretisierten Wertebereichs
`n_iter`: sei die Anzahl der Iterationsschritte bei Erreichen des Abbruchkriteriums.
Die Spalten der Ergebnisdatei sind

```
nx ny  n_iter
```

Erstellen Sie mit Hilfe der `gnuplot` Kommandos aus Übung 3 die Ergebnisbilder.

```
gnuplot> set pm3d map
gnuplot> spl 'ergebnis2A.dat' u 1:2:3 with image
```

Fügen Sie diese in Ihr Dokument ein und dokumentieren Sie die verwendeten Anfangsbedingungen.

Programmeingaben:

1) Wertebereich festlegen: (x_0, y_0) bis (x_M, y_M): 4 Fließkommazahlen (Datentyp double)
   $z_{ll} = x_0 + iy_0$ und $z_{ur} = x_M + iy_M$
2) Unterteilung festlegen: N_{xmax}, N_{ymax}: Anzahl der Intervalle in real bzw. entlang imag Achse
3) Exponent (Ganzzahl): n
4) Maximale Anzahl der Iterationen festlegen: N_{max}
5) Konvergenzradius festlegen: R_c
6) Name der Ergebnisdatei
7) Komplexe Zahl c0:

Iterationsvorschriften + Wertebereiche
Folgende drei Iterationsvorschriften sollen umgesetzt werden:

[1] $z_{i+1} = z_i^2 + c$

a. Wert c wird festgelegt (Eingabedatei: komplexe Konstante c0)
b. Startwert für $z_0 = n_x \delta x + i\, n_y \delta y$ ist Wert aus dem oben angegebenen Wertebereich
c. Ergebnisdatei: Format pro Zeile für Ausgabe als Bild unter gnuplot
$n_x \quad n_y \quad n_{iter}$

Beispiel 1A: (Datei: start1A.dat)

Nummer der Iterationsvorschrift	1		
Wertebereich	(-1.5 , -1.)	(1.5 , 1)	
Unterteilung (Nxmax,Nymax)	750	500	
Exponent	2		
Nmax	2000		
Rc	100		
Dateiname	ergebnis1A.dat		
Komplexe Konstante c0	(-0.75,0.1)		

Beispiel 1B: (Datei: start1B.dat)

Nummer der Iterationsvorschrift	1		
Wertebereich	(-1.5 , -1.)	(1.5 , 1)	
Unterteilung (Nxmax,Nymax)	750	500	
Exponent	2		
Nmax	2000		
Rc	100		
Dateiname	ergebnis1B.dat		
Komplexe Konstante c0	(-0.75,0.55)		

[2] $z_{i+1} = z_i^2 + c$

a. Wert $c = n_x \delta x + i\, n_y \delta y$ ist aus dem oben angegebenen Wertebereich
b. Startwert für z_0 ist (0.,0.) (die können sie im Programm fest einbauen oder über den Übergabeparameter c0 umsetzen)
c. Ergebnisdatei:
   $n_x \quad n_y \quad n_{iter}$

Beispiel 2A: (Datei: start2A.dat)

Nummer der Iterationsvorschrift	2		
Wertebereich	(-2 , -1.)	(1 , 1)	
Unterteilung (Nxmax,Nymax)	750	500	
Exponent	2		
Nmax	200		
Rc	2		
Dateiname	ergebnis2A.dat		
Komplexe Konstante c0	(0.,0.)		

Beispiel 2B: (Datei start2B.dat)

Nummer der Iterationsvorschrift	2		
Wertebereich	(-1.5 , 1.)	(0.,0.)	
Unterteilung (Nxmax,Nymax)	750	500	
Exponent	2		
Nmax	200		
Rc	2		
Dateiname	ergebnis2B.dat		
Komplexe Konstante c0	(0.,0.)		

[3] $z_{i+1} = z_i^4 + c$

a. Wert $c = n_x \delta x + i\, n_y \delta y$ ist aus dem oben angegebenen Wertebereich
b. Startwert für z_0 ist (0,0)
c. Ergebnisdatei:
   $n_x \quad n_y \quad n_{iter}$

Beispiel 2B: (Datei start3A.dat)

Nummer der Iterationsvorschrift	3		
Wertebereich	(-1.5 , -1.)	(1.5 , 1)	
Unterteilung (Nxmax,Nymax)	750	500	
Exponent	4		
Nmax	2000		
Rc	200		
Dateiname	Ergebnis3A.dat		
Komplexe Konstante c0	(0.,0.)		

Tipps: nutzen Sie Funktionen und Funktionszeiger, um Codewiederholungen zu reduzieren.

Viel Erfolg!